

Amendments to the claims,

Listing of all claims pursuant to 37 CFR 1.121(c)

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently amended) A method for performing database operations on data obtained from a web service, the method comprising:

creating at least one proxy table in a database, each proxy table mapping to a method of the web service, wherein said at least one proxy table is automatically created based on a Web Services Description Language (WSDL) description of the web service;

automatically generating meta data about the mapping and storing the meta data in a database table of the database;

creating a shadow table in the database associated with the web service;

in response to a database operation on a particular proxy table, using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping, wherein a row is inserted into the shadow table in response to a database operation including an insert on the particular proxy table;

invoking the particular method of the web service using data in the shadow table as input parameters for invoking the web service;

converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping;

performing the database operation on the data at the database to generate a result set; and

returning the result set in response to the database operation.

2. (Original) The method of claim 1, wherein the web service comprises a service remotely available via a network.

3. (Original) The method of claim 1, wherein the web service has a Web Services Description Language (WSDL) interface.

4. (Previously presented) The method of claim 1, wherein said creating step includes automatically creating said at least one proxy table in response to user input of a WSDL file name of the web service.

5. (Previously presented) The method of claim 4, wherein said step of automatically creating said at least one proxy table includes substeps of:

obtaining the WSDL file the web service; and

automatically creating said at least one proxy table based upon the WSDL file obtained from the web service.

6. (Previously presented) The method of claim 1, wherein said automatically generating step includes automatically generating meta data identifying a particular method of the web service to be invoked when a database operation is received on a particular proxy table.

7. (Original) The method of claim 1, wherein said creating step includes mapping arguments of the method to fields of the proxy table.

8. (Original) The method of claim 1, wherein said creating step includes mapping arguments of the method to equivalent database data types.

9. (Original) The method of claim 1, wherein said creating step includes creating an object encapsulating the mapping of a web method to the database.

10. (Previously presented) The method of claim 1, wherein said automatically generating step includes storing meta data about the mapping between said at least one proxy table and methods of the web service in a system table of the database.

11. (Original) The method of claim 10, wherein said step of converting results includes consulting the mapping for converting the results into data for application at the database.

12. (Previously presented) The method of claim 1, wherein the database operation includes a JOIN operation and said step of performing the database operation includes joining data obtained from invoking the particular method of the web service with data stored in the database in generating the result set.

13. (Currently amended) The method of claim 1, wherein the step of converting results obtained from invoking the web service includes caching the results in the ~~operation includes an insert operation on a given proxy table representing a particular method of the web service and said step of performing the database operation includes inserting a row into a shadow table of the database associated with the web service.~~

14. (Currently amended) The method of claim 1 ~~claim 13~~, wherein said step of performing the database operation includes invoking the particular method of the web service associated with the given proxy table when a column that is an output parameter of the particular method is requested.

15. (Original) The method of claim 1, wherein said step of converting the database operation includes creating a Simple Object Access Protocol (SOAP) request for invoking the particular method of the web service.

16. (Original) The method of claim 15, wherein said step of invoking the particular method includes transmitting the SOAP request to a remote web service.

17. (Original) The method of claim 1, wherein said step of invoking the particular method includes receiving results from the web service.

18. (Original) The method of claim 1, wherein said step of converting results includes converting results received in Simple Object Access Protocol (SOAP) format.

19. (Original) The method of claim 1, wherein said step of converting results

includes converting results received in Extensible Markup Language (XML) format.

20. (Original) A computer-readable medium having processor-executable instructions for performing the method of claim 1.

21. (Previously presented) A downloadable set of processor-executable instructions for performing the method of claim 1 stored on a web server.

22. (Currently amended) In a computer connected to a network and having access to a remote service, a system for performing operations at a database on data obtained from the remote service, the system comprising:

a mapping module for creating database tables representing at least some methods of the remote service accessed through a defined interface and storing mapping data regarding methods of the remote service in a database system table, wherein said database tables and said mapping data are automatically created based on a Web Services Description Language (WSDL) interface of the remote service;

at least one shadow table in the database associated with the remote service;

an invocation module for converting a database operation on a database table representing a method of the remote service into a call for invoking the method using the mapping data, wherein said invocation module inserts a row into the shadow table when an operation including an insert is received on a given database table representing a method of the web service and subsequently uses data in the shadow table to provide input parameters for invoking the method of the web service;

a communication module for transmitting the call for invoking the method to the remote service, and returning result values from invoking the method to the database; and

a conversion module for converting result values received from the method into database format, performing the database operation on the converted result values to generate a database result set, and returning the database result set in response to the database operation.

23. (Original) The system of claim 22, wherein the remote service comprises an

application available via a network.

24. (Previously presented) The system of claim 22, wherein said database tables are automatically created in response to user input of a file name of the Web Services Description Language (WSDL) interface.

25. (Previously presented) The system of claim 24, wherein said mapping module creates a shadow table in the database associated with the web service.

26. (Previously presented) The system of claim 25, wherein said invocation module inserts a row into the shadow table associated with the web service a when an operation including an insert is received on a given database table representing a method of the web service.

27. (Previously presented) The system of claim 25, wherein said invocation module invokes the method of the web service associated with the given database table when a column that is an output parameter of the method is requested.

28. (Previously presented) The system of claim 26, wherein said invocation module uses data in the shadow table to provide input arguments for invoking the method of the web service.

29. (Original) The system of claim 22, wherein said mapping module creates an object encapsulating the mapping of a method of the remote service to a database table.

30. (Previously presented) The system of claim 22, further comprising:
a mapping repository for storing mapping data regarding mappings between database tables and methods of the remote service in the database system table.

31. (Original) The system of claim 30, wherein the conversion module consults the mapping repository for converting result values into database format.

32. (Previously presented) The system of claim 22, wherein the operation received on the database table comprises a JOIN operation and said conversion module joins result values obtained from invoking the method with data stored in the database.

33. (Original) The system of claim 22, wherein said invocation module binds the data from the operation to a Simple Object Access Protocol (SOAP) call for invoking the method of the remote service.

34. (Original) The system of claim 22, wherein said invocation module converts data from the database operation into Extensible Markup Language (XML) format.

35. (Original) The system of claim 22, wherein said invocation module creates a Simple Object Access Protocol (SOAP) request for invoking the method of the remote service.

36. (Original) The system of claim 35, wherein said communication module sends the SOAP request to the remote service.

37. (Original) The system of claim 22, wherein said conversion module converts result values received in Simple Object Access Protocol (SOAP) format into database data types.

38. (Original) The system of claim 22, wherein said conversion module converts result values received in Extensible Markup Language (XML) format into database data types.

39. (Previously presented) The system of claim 22, wherein said database operation received by the invocation module comprises a database query received from a user and said conversion module returns a database result set to the user in response to said database query.

40. (Currently amended) In a database system, a method for performing database queries on data available from an application, the method comprising:

establishing communication between a database and an application having an interface;

creating database tables to represent at least some functions of the application based on the interface, each database table mapping to a corresponding function of the application, wherein said database tables are automatically created based on a Web Services Description Language(WSDL) interface of the application, and each of said database tables having an associated shadow table;

automatically generating meta data about the mapping and storing the meta data in a system table of the database;

in response to a database query received on a database table corresponding to a function of the application that includes an insert operation on the database table,
inserting a row into the shadow table associated with the database table;

generating input arguments expected by the function based on the database query, ~~and~~ the mapping meta data and data from the shadow table;

invoking the function with the input arguments and receiving results from
invoking the function;

converting the results into a database result set; and

returning the database result set in response to the database query.

41. (Currently amended) The method of claim 40, wherein said step of invoking the function includes storing the results in the shadow table associated with the database table~~generating input arguments includes inserting a row into a shadow table of the database associated with the application in response to a database query including an insert operation on a given database table corresponding to a function of the application.~~

42. (Currently amended) The method of claim ~~40~~⁴¹, wherein said step of invoking the function includes invoking the function associated with the given database table when a column that is an output parameter of the function is requested.

43. (Canceled)
44. (Previously presented) The method of claim 40, wherein said automatically generating step includes automatically generating meta data identifying a particular function to be invoked when an operation is received on a given database table.
45. (Original) The method of claim 40, wherein said step of creating database tables includes mapping arguments of a given function to columns of the corresponding database table.
46. (Original) The method of claim 40, wherein said step of invoking the function includes binding data from the database query to a Simple Object Access Protocol (SOAP) call.
47. (Original) The method of claim 40, wherein said step of invoking the function includes converting data from the database query into Extensible Markup Language (XML) format.
48. (Original) The method of claim 40, wherein said step of invoking the function includes creating a Simple Object Access Protocol (SOAP) request for invoking the function.
49. (Original) The method of claim 48, wherein said step of invoking the function includes transmitting the SOAP request to a remote server.
50. (Original) The method of claim 40, wherein said step of invoking the function includes receiving results in Extensible Markup Language (XML) format.
51. (Original) The method of claim 40, wherein said step of invoking the function includes receiving results in Simple Object Access Protocol (SOAP) format.

52. (Original) The method of claim 40, wherein said step of converting the results includes converting results received in Simple Object Access Protocol (SOAP) format.

53. (Original) The method of claim 40, wherein said step of converting the results includes converting results received in Extensible Markup Language (XML) format.

54. (Original) A computer-readable medium having processor-executable instructions for performing the method of claim 40.

55. (Previously presented) A downloadable set of processor-executable instructions for performing the method of claim 40 stored on a web server.